Comparison of index-3, index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral and bilateral constraints

Mounia Haddouni, Vincent Acary, Jean-Daniel Beley

Comparison of index-3, index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral and bilateral constraints

Mounia Haddouni[1,2], Vincent Acary[1], Jean-Daniel Beley[2]

[1] INRIA Bipop team: mounia.haddouni@inria.fr, vincent.acary@inria.fr
[2] ANSYS, France: jean-daniel.beley@ansys.com

Abstract

This work is devoted to the study of some numerical solvers for the numerical time integration of nonsmooth multibody systems with unilateral or bilateral constraints. In the framework of event-detecting time–stepping schemes (a.k.a. event–driven schemes), an index-3 Differential Algebraic Equation (DAE) has to be solved between two events with a constant number of bilateral constraints. In this paper, we compare several solvers for index-1, index-2 and index-3 DAEs in the specific context of unilateral contact and impact. These solvers will be compared in terms of drift of the constraints which is an important feature when we have to update the index sets of the active unilateral constraints. Their implementation and efficiency with fixed and adaptive time–step strategies will also be reviewed. Finally, we discuss the way they handle cases when the constraints are not sufficiently differentiable, which is of utmost importance for practical applications.

Keywords: *nonsmooth mechanics, unilateral contact, impacts, event-driven schemes, constraints, DAE, numerical solvers, LCP, NSCD*

1 Introduction

The simulation of multi-body systems composed of rigid bodies or flexible bodies, such as the one presented in Fig.1, with joints, unilateral contact and Coulomb's friction requires the development of sophisticated numerical schemes (see [1, 2] for a review). In the framework of event-detecting time–stepping schemes (a.k.a. event–driven schemes), an index-3 Differential Algebraic Equation (DAE) has to be solved between two events with a constant number of bilateral constraints[3, 4, 5]. In this paper, we compare several solvers for index-1, index-2 and index-3 DAEs in the specific context of unilateral contact and impact. These solvers will be compared in terms of drift of the constraints, implementation and the way they handle cases when the constraints are not sufficiently differentiable.

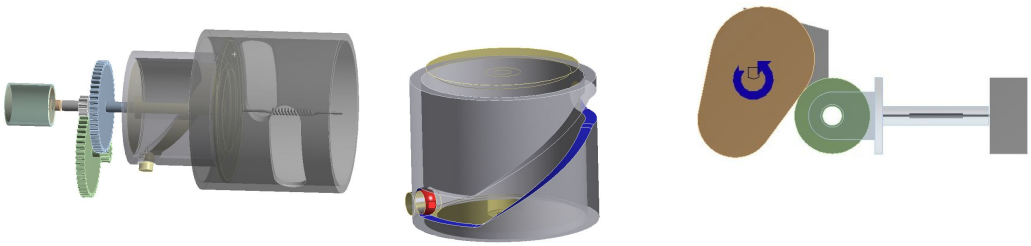


Figure 1: Some illustrations of systems with nonsmooth constraints. A valve system with a zoom on the cylindrical cam and a standard cam–follower system.

The dynamics of a rigid multibody system with m frictionless contact points can be written, using the Lagrangian formalism, as

$$\begin{cases} \dot{q}(t) = v(t) & \text{(1a)} \\ M(q(t))\,\dot{v}(t) = F(t, q(t), v(t)) + G^{\top}(q(t))\lambda(t) & \text{(1b)} \\ 0 \leq \lambda(t) \perp g(q(t)) \geq 0, & \text{(1c)} \end{cases}$$

where $q(t) \in \mathbb{R}^n$ is the generalized coordinates vector, $v(t) \in \mathbb{R}^n$ is the generalized velocities vector, $M(q(t)) \in \mathbb{R}^{n\times n}$ is the symmetric definite positive matrix of inertia, $F(t, q(t), v(t))$ comprises the external applied loads, the non linear

interactions between bodies and the non linear inertial terms, $g(q(t)) \in \mathbb{R}^m$ is the vector of constraints, $G = \nabla_q^\top q(q)$ is the Jacobian of the constraints and $\lambda(t) \in \mathbb{R}^m$ is the Lagrange multiplier which is related to the force at the contact point. The complementarity condition (1c) models the unilateral contact, also termed as the *Signorini condition* at the position level.

We suppose that impacts occur in infinitely short periods so that the displacements of the bodies during the contact period can be negligible and we use the Newton impact law with a coefficient of restitution e. If $g(q(t_i)) = 0$, that is the contact is closed (we say also active) at time t_i, we compute the impulse p_i and the post-impact velocity $v^+(t_i)$ by solving the impact equations

$$\left\{\begin{array}{l} M(q(t_i))(v^+(t_i) - v^-(t_i)) = G^\top(q(t_i))p_i \\ U_{\text{N}}^+(t_i) = G(q(t_i))v^+(t_i) \\ U_{\text{N}}^-(t_i) = G(q(t_i))v^-(t_i) \\ 0 \leq U_{\text{N}}^+(t_i) + e\,U_{\text{N}}^-(t_i) \perp p_i \geq 0, \end{array}\right. \tag{2}$$

where $U_{\text{N}}(t) = \dot{g}(q(t)) = G(q(t))\dot{q}(t)$ is the relative normal velocity at contact. The complementarity condition in (2) describes the Signorini condition written at the velocity level and augmented of the impact law.

When an event-driven strategy is chosen for the numerical time integration, a solver for differential algebraic equations has to be used between two events for solving the smooth system with equality constraints $g^\alpha(q) = 0, \alpha \in I_2$. The index set I_2 belongs to the family of index sets generally introduced in order to characterize the state of the contacts:

- the index set I_0 of all possible constraints in the system $I_0 = \{1, ..., m\}$,
- the index set I_1 of contacts activated in position: $I_1 = \{i \in I_0 | g_i(q(t)) = 0\}$,
- the index set I_2 of all closed contacts $I_2 = \{i \in I_1 | \dot{g}_i(q(t)) = 0\}$.

From the numerical point of view and with the assumption that the constraints are smooth enough, the following smooth dynamics,

$$\left\{\begin{array}{ll} \dot{q}(t) = v(t) & \text{(3a)} \\ M(q(t))\dot{v}(t) = F(t, q(t), v(t)) + G^\top(t, q(t))\lambda & \text{(3b)} \\ g^\alpha(q(t)) = 0,\ \lambda^\alpha \geq 0, \quad \alpha \in I_2. & \text{(3c)} \end{array}\right.$$

is integrated with a DAE solver, during periods of time over which the index set I_2 is constant. An event detection is performed to find the time where new contacts occur or some contacts need to be released. When an event is detected, the impact equations (2) are solved and the state of the dynamical system is updated. A complete algorithm for the event-driven schemes is detailed in [1].

It is well known that index-3 DAEs are difficult to numerically handle [3, 6]. Therefore the dynamics is usually integrated as an Ordinary Differential Equation (ODE) by reducing the original index of the system from 3 to 1. It amounts to solving the problem at the acceleration level by differentiating twice the constraints. Roughly, index reduction consists in differentiating the constraint as many times as necessary to get a set of equations that may be solved using methods for lower index problems. Hence, if the constraint g is differentiated once with respect to time, one obtains an index-2 DAE, we solve in this case (3a) and (3b) together with

$$G(t, q(t))v(t) = 0. \tag{4}$$

If g is twice differentiated, one gets an index-1 DAE composed of (3a) and (3b) with

$$G(t, q(t))\dot{v}(t) + G_{qq}(v, v) = 0. \tag{5}$$

Numerically, the index reduction leads to the phenomenon of drift of the constraints, at different levels depending on the index. Indeed, as opposed to the continuous time case, enforcing (5) in discrete time does not imply that (3c) and (4) are satisfied.

2 Presentation of the chosen time-integration schemes

2.1 An index–3 DAE solver: the generalized-α–scheme

This scheme dedicated to the integration of index-3 DAEs (3) has been studied by O. Brüls and M. Arnold [6]. It is an adaptation of the standard α–scheme for the resolution of index-1 DAEs in the context of computational Mechanics of solids. Let us introduce the acceleration-like variable a defined by the recurrence relation

$$(1 - \alpha_m)a_{n+1} + \alpha_m a_n = (1 - \alpha_f)\ddot{q}_{n+1} + \alpha_f \ddot{q}_n. \tag{6}$$

The discretization of equation (3) is given by

$$\begin{cases} q_{n+1} = q_n + h\dot{q}_n + h^2(\frac{1}{2} - \beta)a_n + h^2\beta a_{n+1} \\ \dot{q}_{n+1} = \dot{q}_n + h(1-\gamma) + h\gamma a_{n+1}, \end{cases} \tag{7}$$

where the constants α, α_m, β and γ should be chosen suitably so that the scheme is stable. In [6], it is said that the algorithm is unconditionally stable if the coefficients are chosen such that for $\rho_\infty < 1$

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} < \frac{1}{2}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} < \frac{1}{2}, \quad \gamma = \frac{1}{2} + \alpha_f - \alpha_m > \frac{1}{2}, \quad \beta = \frac{1}{4}(\gamma + \frac{1}{2})^2. \tag{8}$$

The scheme is based on a prediction step and a correction step where some Newton iterations are performed in order to reduce the dynamical and the constraint residuals defined by

$$\begin{cases} R_q = M(q,t)\ddot{q} - F(q,\dot{q},t) - G^\top(q)\lambda \\ R_\lambda = g(q). \end{cases} \tag{9}$$

The Newton iterations amount to solving of the following linear system

$$\begin{bmatrix} \beta^{'} M(q,t) + \gamma^{'} C_t(q,\dot{q},t) + K_t(q,\dot{q},t) & -G^T(q) \\ G(q) & 0 \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} R_q \\ R_\lambda \end{bmatrix}, \tag{10}$$

where $\beta^{'} = \frac{1-\alpha_m}{h^2\beta(1-\alpha_f)}$, $\gamma^{'} = \frac{\gamma}{h\beta}$, M is the inertia matrix, $K_t = \frac{\partial(M\ddot{q} - F + G^\top\lambda)}{\partial q}$ is the stiffness matrix, and $C_t = -\frac{\partial F}{\partial \dot{q}}$ is the damping matrix. Note that this algorithm aims at maintaining the constraints at the position level, but it can also be reformulated to write the constraints at the velocity level or at the acceleration level [7].

For the control of time step size, we compute an estimation of the solution $y = [q_{n+1}, \dot{q}_{n+1}]$ using a 1^{st} order Newmark scheme with $\gamma > 1/2$, and we define the truncation error as

$$\text{err} = \|y_{\alpha-\text{scheme}} - y_{\text{Newmark}}\|. \tag{11}$$

The optimal step size is then given with

$$h_{\text{opt}} = \text{safe}\left(\frac{\text{tol}\ h}{\text{err}}\right)^{1/p} h, \tag{12}$$

where safe is a safety factor, tol the user required tolerance and $p = 2$ is the order of the α–scheme.

2.2 An index–2 DAE solver: the Half-Explicit Method of order 5 (HEM5)

The HEM5 solver is based on a half-explicit 5 order Runge-Kutta(RK) method with 8 stages, described in [8]. Half-Explicit RK methods have been deeply studied by E. Hairer in [9]. The HEM5 solver is suited to the integration of DAEs, reduced to the index-2 formulation. The numerical approximation of the equations of motion is given by

$$\begin{cases} M(Q_i, t_n + c_i h)\dot{V}_i = F(t_n + c_i h, Q_i, V_i) + G^\top(t_n + c_i h, Q_i)\Lambda_i \\ \dot{Q}_i = V_i \\ G(t_n + c_i h, Q_i)V_i = 0, \end{cases} \tag{13}$$

where the stages are evaluated as follows : $Q_i = q_n + h\sum_{j<i} a_{ij}\dot{Q}_j$ and $V_i = v_n + h\sum_{j<i} a_{ij}\dot{V}_j$.

At each stage, the estimations of position and velocity Q_i and V_i are explicitly computed, while Λ_i and $\dot{V}_i$ are implicitly obtained by solving the linear system

$$\begin{bmatrix} M(Q_i, t_n + c_i h) & -G^T(Q_i, t_n + c_i h) \\ G(Q_{i+1}, t_n + c_{i+1} h) & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} F(Q_i, V_i, t_n + c_i h) \\ r_i \end{bmatrix}, \tag{14}$$

where $r_i = -\frac{G(Q_{i+1})}{h a_{i+1,i}}(v_n + h\sum_{j=1}^{i-1} a_{i+1,j}\dot{V}_j)$. At the end of the time step, the numerical solution is given by $q_{n+1} = Q_9 = q_n + h\sum_{i=1}^{8} b_i\dot{Q}_i$ and $v_{n+1} = V_9 = v_n + h\sum_{i=1}^{8} b_i\dot{V}_i$, with $b_i = a_{9i}$. The computation of coefficients c_i and a_{ij} is detailed in [8]. It is worthwhile to note that the HEM5 solver is constructed in a way that the constraint at the velocity

level is satisfied at each stage: $G(Q_i)V_i = 0\,, \forall i = 1 \dots 8$. In order to get the acceleration $\dot{v}$ and the Lagrange multiplier λ at the end of the time step, an additional linear system has to be solved

$$\begin{bmatrix} M & -G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ r \end{bmatrix} \quad \text{with } r = -G_{qq}(v,v). \tag{15}$$

Concerning the HEM5 solver, V. Brasey and E. Hairer [8] define an error based on the 7^{th} and 8^{th} estimations Q_7 and Q_8 such that

$$\begin{aligned} \mathrm{err}_1 &= \|q_{n+1} - Q_8\|_s = O(h^4) \\ \mathrm{err}_2 &= \|q_{n+1} - q_n - h(\tfrac{5}{2}Q_7 - \tfrac{3}{2}Q_8)\|_s = O(h^3). \end{aligned} \tag{16}$$

Finally, the practical error estimation is computed by

$$\mathrm{err} = \frac{{\mathrm{err}_1}^2}{\mathrm{err}_1 + 0.01\,\mathrm{err}_2} = O(h^5). \tag{17}$$

The optimal step size is computed with the same rule (12) with $p = 5$.

2.3 An index–1 DAE solver: the Runge–Kutta–Fehlberg of order 4 (RKF45) method

The RKF45 is a 4 order method suited for the integration of ODEs. It belongs to the family of RK embedded methods and provides 2 approximations of the solution to evaluate the error. The scheme is based on 6 estimations of the derivatives defined as $Y_i = y_n + h\, f(\sum_{j<i} a_{ij}\dot{Y}_j)$. In our case, we define $Y = \mathrm{col}(\ q, \dot{q}\ )$ and

$$f = \begin{bmatrix} v \\ -M^{-1}G^\top (GM^{-1}G^\top)^{-1}((Gv)_q v + GM^{-1}F) + M^{-1}F \end{bmatrix}.$$

The 4^{th} order approximation of the solution at the end of the step is computed by $y_{n+1} = y_n + \sum_{i\leq 6} b_i \dot{Y}_j$. A 5^{th} order estimation, used for the computation of the error, is given with $\tilde{y}_{n+1} = y_n + \sum_{i\leq 6} \tilde{b}_i \dot{Y}_j$. The computational error is defined with $\mathrm{err} = \|\tilde{y}_{n+1} - y_{n+1}\|$, and the computation of the optimal step size is given with (12) with $p = 4$.

3 Numerical tests

In this section, numerical tests with different values of the tolerance on the truncation error will be performed on two mechanisms, using the time step control strategy presented in section 2. The aim of these tests is to compare the selected solvers in terms of computational effort and drift of the constraints. Table 1 presents the parameters of time-step control strategy. We set the tolerance on the drift of the constraints to a large value (10^{-3}) in order to prevent from a too severe requirement for some schemes when applying the time-step control strategy.

Table 1: Parameters for time step control

Absolute tolerance (tol)	Minimum time step	Tolerance of Newton's loop	Maximum drift of g and $\dot{g}$	safety factor (safe)
$[10^{-10}, 10^{-2}]$(*)	10^{-6}s	10^{-10}	10^{-3}	0.9

(*) We vary the value of tol to compare the computational effort and the drift of the constraints

Table 2 summarizes the computational effort for a single time step for each solver.

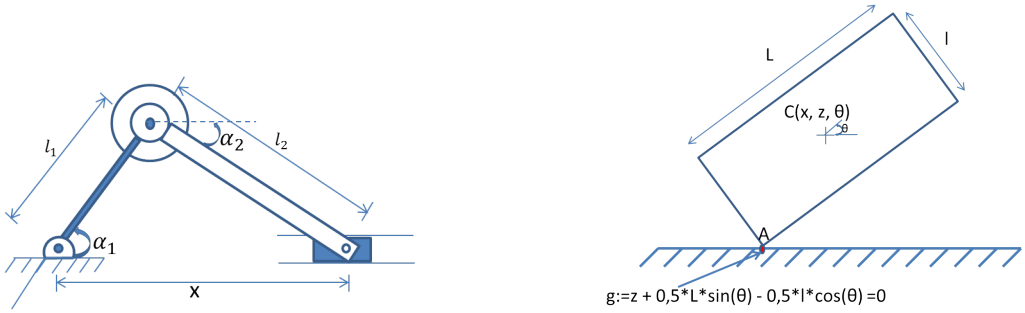
Table 2: Computational effort of the solvers during a single time step

Solver	F	G	G_q	K_t	C_t	# Linear equations / Matrices inversions	Work per time step
HEM5	8	8	1	-	-	9	26
RKF45	6	6	6	-	-	12	30
Newmark scheme	# Newton iterations	# Newton iterations	0	# Newton iterations	# Newton iterations	# Newton iterations	$5\times$ # Newton iterations
α–scheme	# Newton iterations	# Newton iterations	0	# Newton iterations	# Newton iterations	# Newton iterations	$5\times$ # Newton iterations + Newmark's work

3.1 Description of the benchmarks

3.1.1 The slider-crank system

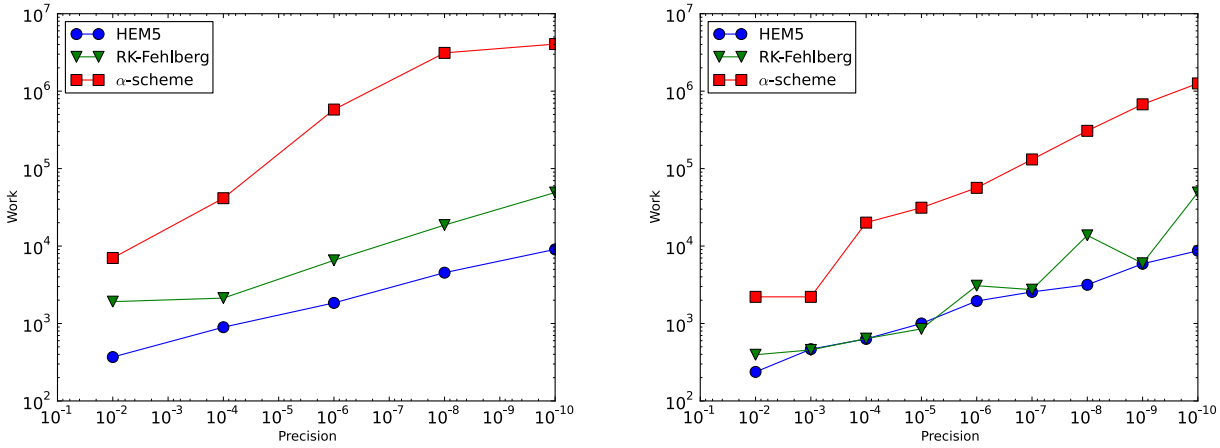
The slider crank system is depicted in Figure 2a.



(a) The slider-crank system

(b) The rocking block

Figure 2: Two academic mechanisms.



(a) Work/Precision diagram of the slider crank

(b) Work/Precision diagram of the rocking block

Figure 3: Work/Precision diagrams for the Slider crank and the Rocking block

Let m_1, m_2 and l_1, l_2 denote the masses and the lengths of the rods, and m_3 the mass of the slider. The system is described with the vector of generalized coordinates $q = [\alpha_1,\, \alpha_2,\, x]^\top$ as defined on Figure 2a. The revolute joint and the

Table 3: Slider-crank: average time step size, work and drift of the constraints for different tolerances

(a) HEM5 solver

tolerance	average h	work	$g_{1_{max}}$	$g_{2_{max}}$	$\dot{g}_{1_{max}}$	$\dot{g}_{2_{max}}$
10^{-2}	0.117	364	$9.63\ 10^{-6}$	$1.24\ 10^{-5}$	$1.02\ 10^{-15}$	$4.65\ 10^{-16}$
10^{-4}	0.061	884	$8.18\ 10^{-8}$	$4.45\ 10^{-7}$	$1.34\ 10^{-15}$	$5.87\ 10^{-16}$
10^{-6}	0.027	1820	$2.27\ 10^{-9}$	$1.20\ 10^{-8}$	$1.02\ 10^{-15}$	$4.70\ 10^{-16}$
10^{-8}	0.011	4472	$1.33\ 10^{-10}$	$4.85\ 10^{-10}$	$1.24\ 10^{-15}$	$4.93\ 10^{-16}$
10^{-10}	$5.25\ 10^{-3}$	8944	$7.80\ 10^{-12}$	$2.91\ 10^{-11}$	$1.33\ 10^{-15}$	$6.27\ 10^{-16}$

(b) RKF45 method

tolerance	average h	work	$g_{1_{max}}$	$g_{2_{max}}$	$\dot{g}_{1_{max}}$	$\dot{g}_{2_{max}}$
10^{-2}	0.037	1890	$2.00\ 10^{-4}$	$9.39\ 10^{-4}$	$2.29\ 10^{-4}$	$9.89\ 10^{-4}$
10^{-4}	0.036	2100	$1.22\ 10^{-4}$	$8.97\ 10^{-4}$	$1.50\ 10^{-4}$	$6.73\ 10^{-4}$
10^{-6}	0.021	6450	$1.44\ 10^{-6}$	$1.21\ 10^{-6}$	$1.10\ 10^{-6}$	$3.94\ 10^{-6}$
10^{-8}	$7.50\ 10^{-3}$	18420	$1.67\ 10^{-8}$	$1.53\ 10^{-7}$	$1.25\ 10^{-8}$	$1.19\ 10^{-7}$
10^{-10}	$2.85\ 10^{-3}$	48480	$4.10\ 10^{-10}$	$4.99\ 10^{-9}$	$8.54\ 10^{-10}$	$4.49\ 10^{-9}$

(c) α–scheme

tolerance	average h	work	$g_{1_{max}}$	$g_{2_{max}}$	$\dot{g}_{1_{max}}$	$\dot{g}_{2_{max}}$
10^{-2}	$7.16\ 10^{-3}$	6915	$2.26\ 10^{-16}$	$4.44\ 10^{-16}$	$8.88\ 10^{-4}$	$9.76\ 10^{-4}$
10^{-4}	$1.79\ 10^{-3}$	40925	$2.78\ 10^{-16}$	$6.66\ 10^{-16}$	$9.83\ 10^{-4}$	$9.98\ 10^{-4}$
10^{-6}	$1.71\ 10^{-4}$	570895	$3.33\ 10^{-16}$	$6.66\ 10^{-16}$	$5.78\ 10^{-5}$	$3.74\ 10^{-4}$
10^{-8}	$2.90\ 10^{-5}$	3077300	$2.22\ 10^{-16}$	$6.66\ 10^{-16}$	$1.83\ 10^{-6}$	$3.54\ 10^{-6}$
10^{-10}	$1.00\ 10^{-6}$	4000000	$3.33\ 10^{-16}$	$6.66\ 10^{-16}$	$5.32\ 10^{-8}$	$8.36\ 10^{-8}$

prismatic joint lead to two constraints which are written as:

$$\begin{cases} g_1(q) = l_1 \sin(\alpha_1) + l_2 \sin(\alpha_2) = 0 \\ g_2(q) = x - l_1 \cos(\alpha_1) - l_2 \cos(\alpha_2) = 0 \end{cases} \tag{18}$$

The parameters of the simulation are: $q_0 = [0, 0, 3]^\top$, $v_0 = [8, -4, 0]^\top$, $m_1 = 1\,\text{kg}$, $m_2 = 1\,\text{kg}$, $m_3 = 1\,\text{kg}$, $l_1 = 1\,\text{m}$, $l_1 = 2\,\text{m}$, $gr = 9.81\,\text{m.s}^{-2}$ (Earth gravity), $t_0 = 0\,\text{s}$ and $T_{end} = 2\,\text{s}$.

3.1.2 The rocking block

The aim is to study the motion of the center of mass of the block while maintaining an edge of the block in contact with the ground, as depicted in Figure 2b. The motion of the block can be described by the generalized coordinates $q = [x,\ z,\ \theta]^\top$. The characteristics of the simulation are: $q_0 = [0.5L\cos(\frac{\pi}{3}) - 0.5l\sin(\frac{\pi}{3}), 0.5L\sin(\frac{\pi}{3}) + 0.5l\cos(\frac{\pi}{3}), \frac{\pi}{3}]^\top$, $v_0 = [0, 0, 0]^\top$, $m_1 = 1\,\text{kg}$, $l = 0.5, \text{m}$, $L = 1\,\text{m}$, $gr = 9.81\,\text{m.s}^{-2}$ (Earth gravity), $t_0 = 0\,\text{s}$ and $T_{end} = 0.5\,\text{s}$. During the simulation time, the point A of the block should stay in contact with the ground on a nontrivial time–interval. This constraint in A is formulated as

$$g(q) = z - 0.5\,L\sin(\theta) - 0.5\,l\cos(\theta) = 0. \tag{19}$$

3.2 Results

3.2.1 Computational effort

In Figures 3a and 3b, we present the precision-work diagrams for the two mechanisms. From the computational effort point of view, Figure 3 shows a big gap between the α–scheme on the one hand and the HEM5 solver and the RKF45 scheme on the other hand. The α–scheme solver is almost 9 times more expensive than HEM5 and 3 times more than the RKF45 scheme. We can say that overall the HEM5 solver is the most computationally efficient since the tolerances on the integration error and on the drift of the constraints are respected with smaller time step sizes.

3.2.2 Drift of the constraints

For each value of the tolerance in the range $[10^{-10}, 10^{-2}]$, we computed the maximum drift of the constraints at both position and velocity levels, for each mechanism and each scheme. Tables 3 and 4 show the results for the slider-crank

Table 4: Rocking block: average time step size, work and maximum drift

(a) HEM5 solver

tolerance	average h	work	g_{max}	$\dot{g}_{max}$
10^{-2}	0.336	234	$3,47\ 10^{-5}$	$4,81\ 10^{-16}$
10^{-3}	0.260	460	$1,13\ 10^{-7}$	$3,34\ 10^{-16}$
10^{-4}	0.179	624	$6,34\ 10^{-8}$	$6,72\ 10^{-16}$
10^{-5}	0.104	988	$3,16\ 10^{-9}$	$6,97\ 10^{-16}$
10^{-6}	0.056	1924	$4,76\ 10^{-10}$	$4,58\ 10^{-16}$
10^{-7}	0.054	2524	$3,78\ 10^{-10}$	$6,45\ 10^{-16}$
10^{-8}	0.046	3120	$6.00\ 10^{-11}$	$6,13\ 10^{-16}$
10^{-9}	0.026	5824	$8,01\ 10^{-11}$	$6,84\ 10^{-16}$
10^{-10}	0.018	8580	$1,29\ 10^{-11}$	$7,14\ 10^{-16}$

(b) RKF45 method

tolerance	average h	work	g_{max}	$\dot{g}_{max}$
10^{-2}	$0,065$	390	$1,70\ 10^{-4}$	$9,50\ 10^{-5}$
10^{-3}	$0,063$	450	$2,21\ 10^{-5}$	$8,07\ 10^{-5}$
10^{-4}	$0,051$	630	$1,21\ 10^{-5}$	$8,07\ 10^{-5}$
10^{-5}	$0,039$	840	$2,02\ 10^{-6}$	$2,38\ 10^{-6}$
10^{-6}	$0,0009$	3036	$5,62\ 10^{-7}$	$9,98\ 10^{-7}$
10^{-7}	$0,017$	2700	$4,93\ 10^{-8}$	$9,75\ 10^{-8}$
10^{-8}	$0,0029$	13560	$7,13\ 10^{-9}$	$1.00\ 10^{-8}$
10^{-9}	$0,0008$	5910	$4,17\ 10^{-10}$	$9,96\ 10^{-10}$
10^{-10}	$7,6\ 10^{-4}$	48510	$5,30\ 10^{-11}$	$1.00\ 10^{-10}$

(c) α–scheme

tolerance	average h	work	g_{max}	$\dot{g}_{max}$
10^{-2}	$7.11\ 10^{-3}$	2180	$8.33\ 10^{-17}$	$9.33\ 10^{-5}$
10^{-3}	$7.11\ 10^{-3}$	2180	$8.33\ 10^{-17}$	$9.33\ 10^{-5}$
10^{-4}	$1.26\ 10^{-3}$	19815	$8.33\ 10^{-17}$	$1.43\ 10^{-3}$
10^{-5}	$6.52\ 10^{-4}$	30870	$8.33\ 10^{-17}$	$2.28\ 10^{-4}$
10^{-6}	$3.12\ 10^{-4}$	55725	$8.33\ 10^{-17}$	$2.79\ 10^{-5}$
10^{-7}	$1.28\ 10^{-4}$	129555	$8.33\ 10^{-17}$	$5.77\ 10^{-6}$
10^{-8}	$4.86\ 10^{-5}$	303435	$8.33\ 10^{-17}$	$4.12\ 10^{-7}$
10^{-9}	$1.97\ 10^{-5}$	667055	$8.33\ 10^{-17}$	$5.30\ 10^{-8}$
10^{-10}	$6.48\ 10^{-6}$	1241790	$1.11\ 10^{-16}$	$7.50\ 10^{-9}$

mechanism and for the rocking block.

Slider crank: With the HEM5 solver, the drift of the constraints at the velocity level vanishes (at the machine accuracy). The drift at the position level is acceptable even for large tolerances on the integration error. With the RKF45 method, the tolerance at the integration error is met with large time step sizes ($\simeq 10^{-2}$, 2^{nd} column of Table 3b) but the drift of the constraints is much higher than with the HEM5 solver. For the slider crank mechanism, the drift of g_1 and g_2 (columns 4 and 5 of Table 3c) using the α–scheme, vanishes (at the machine accuracy). The error at the velocity level is quite large (one order less than the time step size). If the magnitude of this drift exceeds the tolerance set for the index set I_2, then the evaluation of this index set will be wrong or difficult to perform.

Rocking block: From Tables 4a, 4b and 4c, we can observe that for the large tolerances, the HEM5 solver and the RKF45 method are equivalent from the computational cost point of view, but this later is more costly for tight tolerances. But both solvers are doing better than the α–scheme. The drift of the constraints at the velocity level for the α–scheme is quite high for large tolerances (large time step sizes) by contrary to the two other solvers. This drift may lead to a wrong estimation of the closed contacts belonging to the set I_2.

As said in the sections describing the HEM5 solver and the α–scheme, the first solver is constructed in a manner that

enables to hold the constraints at the velocity level while the second one maintains the constraints at the position level, and both of them achieve very well these goals. The drift of the constraints using the RKF45 scheme is maintained at acceptable levels depending on the time step size being used. Overall, the HEM5 solver gives the best results with regards to the drift of the constraints at both position and velocity levels, even for large time step sizes.

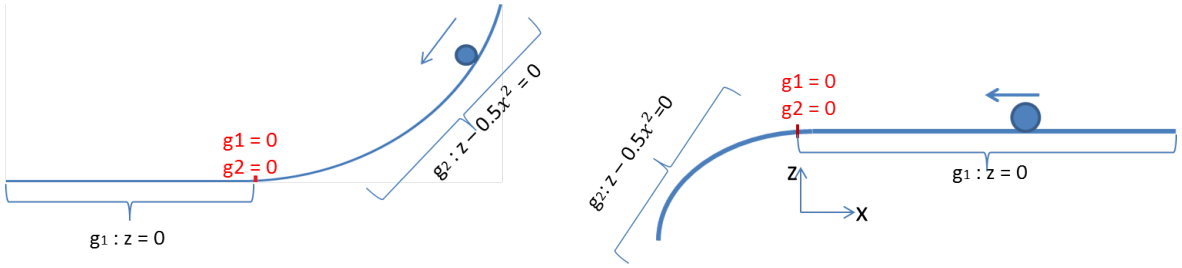
With regards to these results, it seems that discretizing the constraints at the velocity level is a good compromise to reduce their drift at both position and velocity levels.

4 Bodies with geometrical singularities

In this section, we will discuss the changes that occur when there are some edges or other discontinuities in a given geometry. These discontinuities make the integration of the equations of motion difficult, and could even lead to incoherent results if they are not correctly treated. In the case of the HEM5 solver and the RKF45 scheme, these singularities lead to a discontinuity in the term G_{qq} of Equation (5), this leads to a jump in the acceleration and in the contact force. But these discontinuities also make the systems (14) and (10) ill–posed. Let us start with illustrating the problem with some examples. Then, we will propose some solutions to overcome this problem.

4.1 Illustration of the problem

Let us illustrate this with two problems presented in Fig. 4a and Fig. 4b. For the system of Fig. 4a, when the sphere reaches



(a) (b)

Figure 4: Systems with geometrical discontinuities, illustrative examples

the connection, both constraints are active and with the HEM5 solver for example, the system (14) is not well-posed, here is a numerical example of such a system at the discontinuity:

$$\begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 6.65031e^{-5} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_1 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -9.81 \\ 0 \\ -3.62238e^{-2} \\ 4.41971e^{1} \end{bmatrix}$$

At $x = 0$, two constraints are active ($g_1 : z = 0$ and $g_2 : z - x^2 = 0$), but the Hessians of these constraints are not equal at the edge and this leads to a discontinuity in term G_{qq}. In the next section, we propose two methods to solve this problem.

4.2 Solutions to the problem

4.2.1 Solution 1: Formulating the system as a LCP

The first solution consists in re-writing the linear systems (14) and (15) (in the case of HEM5, computation of the estimations of the accelerations and Lagrange multipliers) as a Linear Complementarity Problem (LCP)

$$\begin{cases} M(Q_i)\dot{V}_i - G^\top(Q_i)\Lambda_i = F(Q_i, V_i, t_i) \\ G(Q_{i+1})\dot{V}_i - r_i \geq 0 \\ 0 \leq \dot{V}_i \perp \Lambda_i \geq 0. \end{cases} \tag{20}$$

This formulation transforms the unfeasible linear problem given by system (14) with two equality constraints into a feasible LCP. This transformation enables to find a solution to the problem by relaxing one of the equality constraints at the edge. For our simulations, we used for the resolution of these LCPs Lemke's algorithm available in the open-source

software SICONOS[10][1]. When applied to the examples of Fig. 4a and 4b, Lemke's algorithm provides some estimations of the acceleration and the contact forces which correspond to the expected theoretical ones.

4.2.2 *Solution 2: using the Moreau–Jean's time–stepping scheme*

The 2^{nd} solution consists in using an event–capturing time-stepping scheme, for instance the Moreau–Jean's time–stepping scheme [11, 12]. The event–capturing time-stepping scheme is used here to overcome the difficulty triggered by geometries with non-continuous curvatures. More precisely, it is used to compute the dynamics in the zones containing the geometrical discontinuities. The following pseudo-algorithm explains the way an event–capturing scheme method is used in the whole integration process:

- Detect the time t_d when the discontinuity (edge, undetermined normal) occurs.
- Integrate the dynamics over the time–step $[t_d, t_d + h]$ with the event–capturing scheme.
- Compute the set I_d of active constraints at $t_d + h$.
- Go back at time t_d and solve the dynamics with a DAE solver with taking into account only the constraints belonging to I_d.

We favor the Moreau–Jean method as a solution to the presented problems rather than the LCP (previous subsection) solution since it relies on a mechanical reasoning for finding the correct index set of active constraints on the right of a discontinuity. Thus, the results do depend only on the dynamics of the mechanical problem.

5 Conclusion

This article aimed at comparing index-3, index-2 and index-1 DAE solvers. We chose the generalized-α scheme for the first category, the HEM5 solver for the 2^{nd} one and the RKF45 method for the 3^{rd} one. We wanted to compare the solvers in terms of stabilization of the drift of the constraints, of computational effort and the problems they face in case of geometrical singularities. For this aim, we made simulations on two mechanisms, using fixed time steps and also with adapting the time step size to the user defined tolerance on the truncation error and the maximum drift. After analyzing the time step sizes used by each solver, the computational cost, the maximum drift of the constraints and the behavior in the zones containing singularities, we can draw the following conclusions:

- *Computational effort*. Even though the HEM5 solver contains more stages (8) than the RKF45 scheme (6), the numerical effort of the first solver is lower than that of the second one when a strategy of time step control is used. Indeed, the order (5) of the HEM5 solver in addition to its characteristics of reducing the violation of the constraints, enable to use larger step sizes than those used for the RKF45 scheme, and then to reduce the computational effort. Both HEM5 and RKF45 give better results than the generalized-α scheme that needs very small time step sizes mainly due to its lower order (2) and the drift at the velocity level. But we know that the generalized-α scheme is a low order scheme that is well suited for flexible multibody dynamics where high frequency non-linear dynamics can render the integration difficult. In this latter case, half–explicit methods can be in troubles.
- *Drift of the constraints*. The HEM5 solver enforces the constraints at the velocity level and it enables to reduce drastically the drift at the position level. This enables often to perform the integration without any procedure of projection of the constraints on the admissible manifold. This is not the case of the RKF45 scheme where a projection on the constraints is mandatory in some cases. The generalized-α scheme enforces the constraints at the position level but the drift at the velocity level may lead to numerically losing the contact depending on the tolerances that are chosen for the index set I_2 of closed contacts defined in Section 1. *Discretizing the dynamics with a formulation of the constraint at the velocity level seems to be a good compromise to stabilize the drift at acceptable tolerances.*
- *Geometrical discontinuities*. None of the solvers enable to avoid integration issues (systems ill-posed when constraints are not compatible,...) that occur at some geometrical singularities. We presented two solutions to this problem: formulating the systems that become ill-posed at the singularities as LCPs and solve them with some dedicated LCP algorithm. To this aim, a range of LCP algorithms is proposed in the SICONOS software [10]. This solution gives correct results when the problem is feasible. The second solution is using en event–capturing time–stepping scheme [11, 12] to compute the dynamics in the singularity zone. The last solution is promising since it is only driven by the dynamics and does not depend on any feasibility condition.

[1] `http://siconos.gforge.inria.fr`

In the light of this preliminary study, some further work is planned. The PHEM56 solver of Murua (1995), which is a 6 stages partitioned half-explicit method of order 5 constructed in the same manner as the HEM5 solver, may be used instead of this latter to save the computational effort due to the 2 additional stages in HEM5. In this article, we worked mainly with semi-implicit schemes. Further works will deal with extrapolation solvers such as the MEXX solver by Lubich and al.(1992), and the use of hybrid mixed time stepping integrators in [13] and projected integration schemes in [14].

References

[1] V. Acary and B. Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems*. Springer, 2008.

[2] C. Studer. *Numerics of Unilateral Contacts and Friction. – Modeling and Numerical Time Integration in Non-Smooth Dynamics*, volume 47 of *Lecture Notes in Applied and Computational Mechanics*. Springer Verlag, 2009.

[3] E. Eich-Soellner and C. Führer. *Numerical methods in multibody dynamics*. Teubner, 1998. Reprint Lund university 2002.

[4] R. von Schwerin. *Multibody System Simulation. Numerical Methods, Algorithms and Software*. Springer, 1999.

[5] M. Arnold and W. Schiehlen, editors. *Simulation Techniques in Applied Dynamics*, volume 507 of *CISM Courses and Lectures*. Springer, 2008.

[6] O. Brüls and M. Arnold. Convergence of the generalized-α scheme for constrained mechanical systems. *Multibody System Dynamics*, 2007.

[7] Ch. Lunk and B. Simeon. Solving constrained mechanical systems by the family of Newmark and α–methods. *Zeitschrift für Angewandte Mathematik und Mechanik*, 86(10):772–784, 2006.

[8] V. Brasey. Hem5 user's guide. Technical report, Université de Genève, 1994.

[9] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer, 2002.

[10] V. Acary and F. Pérignon. An introduction to Siconos. Technical Report TR-0340, INRIA, http://hal.inria.fr/inria-00162911/en/, 2007.

[11] J.J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J.J. Moreau and Panagiotopoulos P.D., editors, *Nonsmooth Mechanics and Applications*, pages 1–82. CISM 302, Spinger Verlag, 1988.

[12] M. Jean. The non smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177:235–257, 1999.

[13] V. Acary. Higher order event capturing time–stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts. *Applied Numerical Mathematics*, 62:1259–1275, 2012.

[14] V. Acary. Projected event-capturing time-stepping schemes for nonsmooth mechanical systems with unilateral contact and Coulomb's friction. *Computer Methods in Applied Mechanics and Engineering*, 256:224–250, 2013.